Prezentácia z predmetu
Formálne jazyky a automaty

Marián Šišmiš, 2. ročník, odbor informatika

Zadanie úlohy:

Zostrojte zásobníkový automat A pre jazyk
L1={ucv | u, v ∈ 1{0,1}* a zároveň u, v nie sú binárne zapisy po sebe idúcich čísel takých, že u <= v a "c" je terminalny symbol}
Dokazte spravnost konstrukcie(t.j L(A)=L.)

Inými slovami
L1={ucv| u,v ∈ 1{0,1}* , v≠u+1, t.j. v≠ succ u}. (Ak sa pozeráme na u, v ako na binárne čísla).

Riešenie:

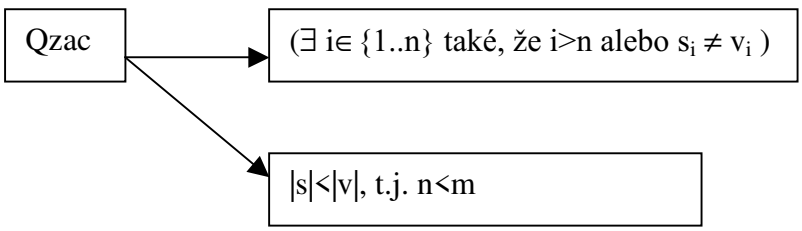
Oznacme L1 jazyk zo zadania. Najskôr si definujeme jazyk L pre, ktory platí, L=L1, pre ktorý zostrojím zásobníkový automat akceptujúci stavom.

Nech platí $u=u_1u_2...u_k$, succ $u=s = s_1s_2...s_n$.(Zrejme n=k alebo n=k+1). Nech platí $v=v_1v_2...v_m$.
u a v začínajú písmenom 1 ⇒ n,m≥1, u1=1,v1=1
L={ucv| u,v∈ 1{0,1}*, (∃ i∈ {1..n} také, že i>n alebo $s_i \neq v_i$) alebo |s|<|v|,t.j. n<m}

L=L1.
Dôkaz:
L1={ucv| u,v ∈ 1{0,1}* , podmienka1}.
L={ucv| u,v∈ 1{0,1}*, podmienka2}.
Zrejme stačí dokázať, že non podmienka1 ⇔ non podmienka2.
⇒: Nech v=u+1 t.j. v=s. potom n=m a pre ∀ i∈ {1..n} platí si=vi a zrejme i≤n.
⇐: Nech |s|<|v| ⇒ s≠v.
Nech ∃ i ∈ {1..n} také, že i>m. Potom n>m ⇒ |s|>|v| ⇒ |s|≠|v|.
Nech ∃ i ∈ {1..n} také, že si>vi Potom |s|≠|v|.

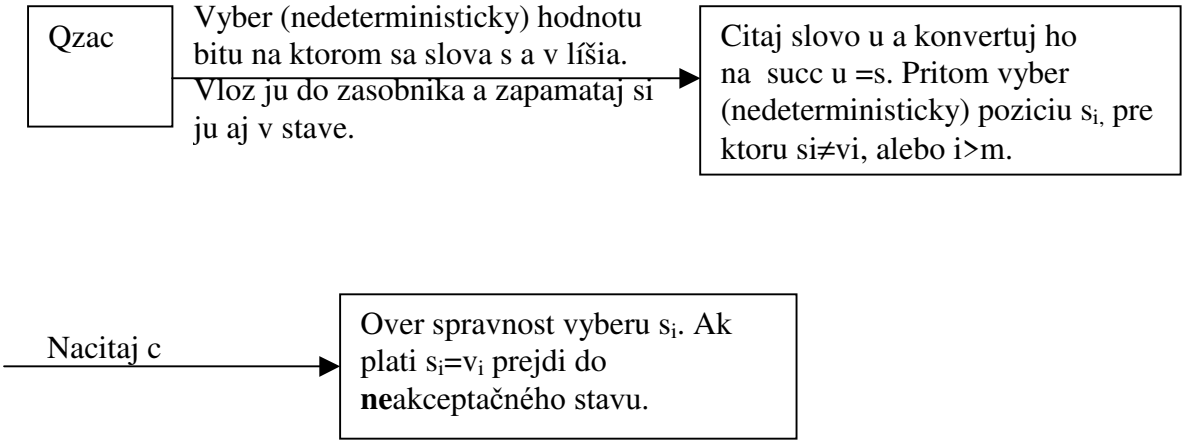
Popis automatu:

Automat rozoznáva 2 prípady uvedené v jazyku L v dvoch od seba nezávislých vetvách.
Na začiatku sa nedeterministicky rozhodne pre tú správnu vetvu. Keďže tieto vetvy sú od seba nezávislé (po vkročení do niektorej vetvy už tá druhá vetva nemá vplyv na výpočet), budem každú vetvu popisovať samostatne.



Prípad 1:

Pri čítaní slova u mohutne využívam nedeterminizmus. Overovanie pri čitaní slova v je v podstate deterministicke.



Konverzia u na succ u:

Nech obsahuje vo svojom zapise 0. Vyberieme poslednu 0, „prepiseme“ ju na 1 a vsetky 1 za ňou prepišeme na 0. – Zrejme je to algoritmu s počítania u+1.

Nech u=11...1, t.j. u neobsahuje 0. Potom u+1 =100...0. Automat to realizuje nasledovne: Zrejme 11...1=011...1. Automat zoberie tuto „virtualnu“ 0, „prepise“ ju 1, a vsetky ostatne jednotky prepise na 0.

(Automat samozrejme nič neprepisuje, len simuluje citanie succ u. Preto v ďalšom texte budem predpokladať, že automat číta succ u, pretože pri čítanie slova u vykonáva vyššie uvedené operácie.)

Vyber a uchovanie pozície porovnávaného bitu, overenie správnosti výberu:

Pri čítaní succ u si do zasobníka za každý prečítaný znak z pásky vložím do zásobníka symbol α. Keď prídem k bitu, ktorý sa rozhodnem porovnávať, vložím si do zásobníka symbol χ. Na označenie, že už som si vybral znak na porovnánie. Pri čítaní slova v postupne mažem zo zásobníka α. Keď ich všetky zmažem, viem že som na tej istej pozícii od začiatku slova a možem porovnať bity.

Na začiatku som ukazal, že je jedno ci počítam pozície od začiatku, alebo od konca. Počítanie pozície od začiatku mi umožní lahšie (v podstate deterministické), overenie správnosti, to znamená zjednodušenie automatu a umožní mi pokryť viac prípadov:

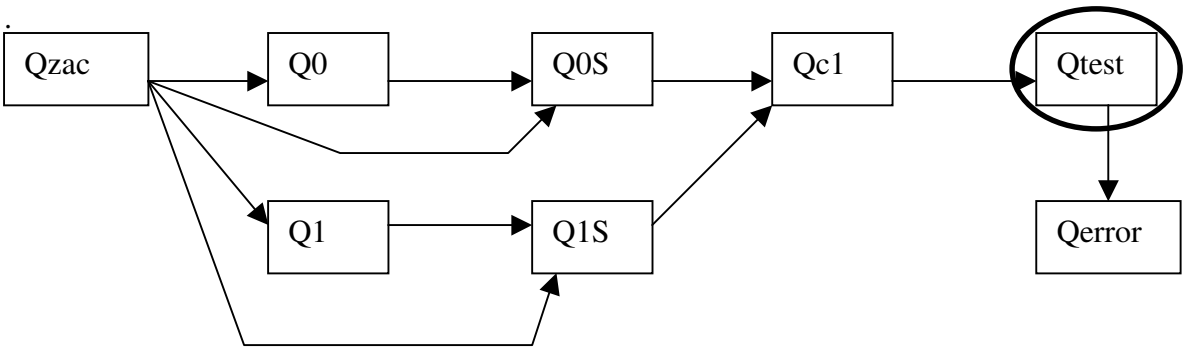
> Overovanie sa vykonáva v akceptačnom stave, z ktorého vyjdem (deterministicky, t.j. nemám inú možnosť), v tom prípade, ak sú porovnávané bity rovnaké. Ak som dočítal slovo v a ešte som sa nedostal k porovnávaniu $\Rightarrow$ vybral som bit s_i, pre ktorý platí i>m to znamená $|s|>|v|$, t. zn. slovo vyhovuje.

Problém zapamätanie si hodnoty:

Riešenie: Uložím hodnotu na spodok zásobníka a potom pri vyberám na porovnávanie len bit s požadovanou hodnotou (preto si zapamätám aj v stave). Keďže mám hodnotu za pamätanú aj v zásobníku, nemusím pri overovaní rozlišovať dve vetvy – vybral som 0, vybral som 1.

Schematický náčrt pomocou stavov

.

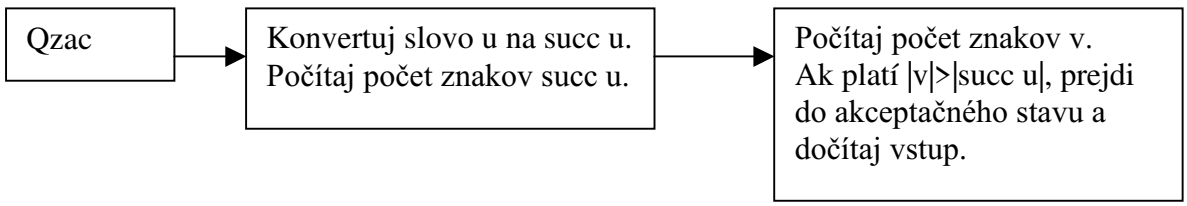


Vysvetlivky:

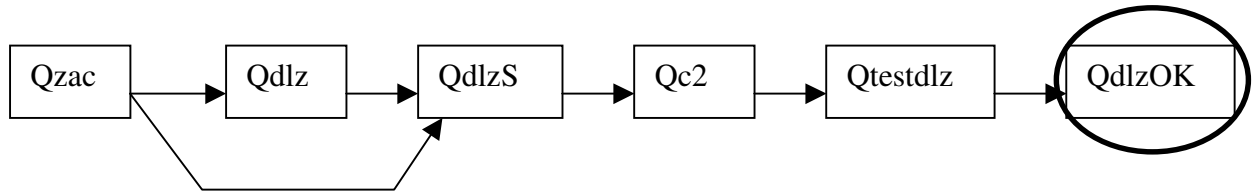
Qzac – začiatočný stav
Q0 – na porovnávanie som vybral 0
Q0S – overenie správnosti generovania succ s – t.j. poslednú 0, prepíšem na 1 a nasledujúce 1 na nuly. V stave Q0S čítam len 1, to znamená ak automat nevybral poslednú 0, zasekne sa.
Q1 – na porovnávanie som vybral 1
Q1S – analogicky s Q0. Dôvod existencie Q0S aj Q1S je v tom, že aj v týchto stavoch môžem vybrať bit na porovnávanie.
Qc1 - nacitanie znaku „c“. Prechod do fázy testovania slova v.
Qtest - deterministické testovanie vybratého bitu.
Qerror – v pripade rovnosti bitov, neakceptačný stav

2. prípad:

Čítam zo vstupu succ u. Do zásobníka si za každý prečítaný znak succ u vložím jeden znak α. Pri čítaní slova v postupne vyhadzujem znaky α zo zásobníka. Keď prídem až na dno zásobníka a ešte je niečo na vstupe $\Rightarrow |v|>|succ\ u|$ a prejdem do akceptačného stavu kde dočítam vstup. Veľkú časť týchto prípadov pokrýva 1. vetva.
Problém je vtedy, keď (succ u =prefix v) & (succ u $\neq$v). Vtedy, ale $|succ\ u|<|v|$ a preto musela byť zavedená aj táto vetva.



Schematický náčrt pomocou stavov:



Qzac - začiatočný stav
Qdlz - počítanie dĺžky succ u
QdlzS - analogicky s Q0S + počítanie dlzky succ u. „Generovanie“ succ u v tomto prípade treba kvôli tomu, že dĺžku u a succ u sa može líšiť o 1.
Qc2 - načítanie znaku „c“ a prechod do fázy počítania dlzky slova v.
Qtestdlz - počítanie dĺžky slova v, ak zistím, že |v|=|succ u| a vstup ešte nie je prázdny – prechod do akceptačného stavu.
QdlzOK - akceptačný stav pre túto vetvu. Dočítanie vstupu.

Vysvetlanie symbolov zásobníka:
Z0 – začiatočný symbol v zásobníku, v 2. vetve dno zásobníka
α - v 1. vetve - počet α v zásobníku – koľko znakov zo succ u sa prečítalo, kým sa vybral bit na porovnávanie
- v 2. vetve - počet α v zásobníku – počet znakov succ u
β_0 – budeme vyberať 0 na porovnávanie
β_1 – budeme vyberať 1 na porovnávanie
C – na vrchu zásobníka – už je vybratý bit na porovnávanie
P – na vrchu zásobníka – slovo je OK, už len dočítať vstup

Formálny zápis automatu A:

$A=(K,\Sigma,\Gamma,\delta,Qzac,Z_0,F)$

$K=\{Qzac,Q0 , Q0S, Q1, Q1S, QC1, Qtest, Qerror, Qdlz, QdlzS, Qc2, Qtestdlz, QdlzOK\}$
$\Sigma=\{0,1,c\}$
$\Gamma=\{Z_0, \beta_0, \beta_1, \alpha, C, P\}$
$F=\{Qtest, QdlzOK\}$

δ-funkcia:
$\delta(Qzac,1,Z_0)=$

1. vetva

$\{(Q0,\beta_0\alpha)$,	budem porovnávať 0, slovo u obsahuje 0
$(Q0S,\beta_0\alpha\alpha)$,	budem porovnávať 0, slovo u neobsahuje 0 (ale succ u bude obsahovať 0), nevyberiem 1. nulu
$(Q0S,\beta_0\alpha C)$	budem porovnávať 0, slovo u neobsahuje 0 (ale succ u bude obsahovať 0), vyberiem 1. nulu
$(Q1,\beta_1\alpha)$,	budem porovnávať 1, slovo u obsahuje 0 nevyberiem 1. jednotku
	budem porovnávať 1, slovo u obsahuje 0 vyberiem 1. jednotku **netreba**, u aj v začínajú 1 takže test by prebehol negatívne

budem porovnávať 1, slovo u neobsahuje 0
vyberiem 1. jednotku **netreba**, u aj v začínajú 1 takže test by prebehol negatívne
budem porovnávať 1, slovo u neobsahuje 0
nevyberiem 1. jednotku **netreba**, lebo potom succ u obsahuje len 0

2. vetva

$(Qdlz,Z_0\alpha)$ slovo u obsahuje 0
$(QdlzS,Z_0\alpha\alpha)$ slovo u neobsahuje 0 (t.j. |succ u|=|u|+1)

1. vetva

$\delta(Q0,0,\alpha)=$ $\{(Q0,\alpha\alpha)$, nie je to posledná 0, nevyber ju
$(Q0,\alpha C)$, nie je to posledná 0, vyber ju
$(Q0S,\alpha\alpha)\}$ je to posledná 0, prepíše sa na 1

$\delta(Q0,0,C)=$ $\{(Q0,C)$, nie je to posledná 0
$(Q0S,C)\}$ je to posledná 0

$\delta(Q0,1,\alpha)=$ $\{(Q0,\alpha\alpha)\}$

$\delta(Q0,1,C)=$ $\{(Q0,C)\}$

$\delta(Q1,0,\alpha)=$ $\{(Q1,\alpha\alpha)$, nie je to posledná 0
$(Q1S,\alpha C)\}$ je to posledná 0, tá sa zmení na 1, musím ju vybrať lebo potom succ u obsahuje len 0

$\delta(Q1,0,C)=$ $\{(Q1,C)$, nie je to posledná 0
$(Q1S,C)\}$ je to posledná 0

$\delta(Q1,1,\alpha)=$ $\{(Q1,\alpha\alpha)$, nevyber túto 1
$(Q0,\alpha C)\}$ vyber túto 1

$\delta(Q1,1,C)=$ $\{Q1,C)\}$

$\delta(Q0S,1,\alpha)=$ $\{(Q0S,\alpha\alpha)$, nevyber túto 0 (v succ u)
$(Q0S,\alpha C)\}$ vyber túto 0 (v succ u)

$\delta(Q0S,1,C)=$ $\{(Q0S,C)\}$

$\delta(Q0S,c,C)=$ $\{(Qc1,\varepsilon)\}$ uc dočítané

$\delta(Q1S,1,C)=$ $\{(Q1S,C)\}$ tu už musí byť 1 vybratá, lebo v succ u sú tu len 0

$\delta(Q1S,c,C)=$ $\{(Qc1,\varepsilon)\}$ uc dočítané

$\delta(Qc1,1,\alpha)=$ $\{(Qtest,\varepsilon)\}$

$\delta(Qtest,0,\alpha)=$ $\{(Qtest,\varepsilon)\}$

$\delta(Qtest,1,\alpha)=$ $\{(Qtest,\varepsilon)\}$

$\delta(Qtest,0,\beta_0)=$ $\{(Qerror,\varepsilon)\}$ rovnaké bity

$\delta(Qtest,0,\beta_1)=$ $\{(Qtest,P)\}$ rôzne bity, slová budú OK

$\delta(Qtest,1,\beta_0)=$ $\{(Qtest,P)\}$ rôzne bity, slová budú OK

$\delta(Qtest,1,\beta_1)=$ $\{(Qerror,P)\}$ rovnaké bity

$\delta(Qtest,0,P)=$ $\{(Qtest,P)\}$ našli sa rôzne bity, už len dočítať vstup

$\delta(Qtest,1,P)=$ $\{(Qtest,P)\}$ našli sa rôzne bity, už len dočítať vstup

2.vetva

$\delta(Qdlz,0,\alpha)=$ {(Qdlz,$\alpha\alpha$), nie je to posledná 0
(QdlzS,$\alpha\alpha$)} je to posledná 0
$\delta(Qdlz,1,\alpha)=$ {(Qdlz,$\alpha\alpha$), nie je to posledná 0
(QdlzS,$\alpha\alpha$)} je to posledná 0

$\delta(QdlzS,1,\alpha)=$ {(QdlzS,$\alpha\alpha$)}
$\delta(QdlzS,c,\alpha)=$ {(Qc2,α)}
$\delta(Qc2,1,\alpha)=$ {(Qtestdlz,ε)}

$\delta(Qtestdlz,0,\alpha)=\{(Qtestdlz,\varepsilon)\}$
$\delta(Qtestdlz,0,Z_0)=\{(QdlzOK,P)\}$
$\delta(Qtestdlz,1,\alpha)=\{(Qtestdlz,\varepsilon)\}$
$\delta(Qtestdlz,1,Z_0)=\{(QdlzOK,P)\}$

$\delta(QdlzOK,0,P)=\{(QdlzOK,P)\}$ Slová sú OK, už len dočítaj vstup.
$\delta(QdlzOK,1,P)=\{(QdlzOK,P)\}$ Slová sú OK, už len dočítaj vstup.

Dôkaz správnosti L(A)=L:

Vzhľadom na podrobný popis konštrukcie automatu a podrobný a okomentovaný popis δ-funkcie dôkaz približne načrtnem. Technické podrobnosti vyplývajú z horeuvedeného popisu.
Zo začiatočného stavu Qzac sa dostanem len načítaním 1, potom čítame len 0 a 1.
Správne „generovanie“ succ u: poslednú 0 prepíšeme na 1, ak nezvolíme poslednú 0, automat sa zasekne v stavoch Q0S, Q1S, QdlzS cez ktoré musí výpočet prejsť (viz schémy).
Potom načítame jedno „c“ a prejdeme do Qc1, prípadne Qc2, znova načítame 1
+ overujeme správnosť podmienok.

Z konštrukcie automatu vyplýva, že že automat akceptuje len slová typu ucv, u,v$\in 1\{0,1\}^*$.
Nech platí $u=u_1u_2...u_k$, succ $u=s = s_1s_2...s_n$.(Zrejme n=k alebo n=k+1). Nech platí $v=v_1v_2...v_m$
$\subseteq$:
Ak je na vstupe slovo ucv, kde succ u=v:
Pre gf$\forall$si, si=vi $\Rightarrow$ V 1. vetve vypočet vždy skončí v stave Qerror.
|succ u|=|v| $\Rightarrow$ V 2.vetve skončí výpočet v stave Qdlz.
$\supseteq$:
Nech w=ucv$\in$ L.
- Nech existuje si$\neq$vi. Zrejme i>1.
Z rozboru δ-funkcie – vieme vybrať na porovnanie ľubovoľný bit okrem prvého $\Rightarrow$ vieme vybrať na porovnanie aj bit si, pri výpočte v 1. vetve skončíme v stave Qtest.

- Nech existuje si. i>m.
i>1 $\Rightarrow$ Vieme ho vybrať $\Rightarrow$ vypočet v 1. vetve skončí v stave Qdlz (viz popis)
- Nech |v|>|succ u|.
Potom výpočet v druhej vetve skončí v stave QdlzOK. (viz popis)